

Application No. 09/668,109

LISTING OF THE CLAIMS

A detailed listing of all claims that are, or were, in the present application, irrespective of whether the claim(s) remains under examination in the application are presented below. The claims are presented in ascending order and each includes one status identifier. Those claims not cancelled or withdrawn but amended by the current amendment utilize the following notations for amendment: 1. deleted matter is shown by strikethrough for six or more characters and double brackets for five or less characters; and 2. added matter is shown by underlining.

1. (Cancelled).
2. (Previously Presented) The system of claim 22, wherein said program directs said processor to convert the HDL coded electronic circuit model portions to binary object code by directing said processor to translate the HDL coded electronic circuit model portions into an intermediate program language code and to compile said intermediate program language code to said binary object code.
3. (Original) The system of claim 2, wherein said intermediate program language code is a C program language code.
4. (Original) The system of claim 3, wherein said C program language code is grouped into code types selected from a group consisting of: evaluation C code and scheduling C code.
5. (Previously Presented) The system of claim 22, wherein said binary object code performs operations that are selected from a group consisting of: initial/always block operations, timing-

Application No. 09/668,109

free procedural operations, task procedural operations, function procedural operations, event control operations, delay control operations, scheduled procedural operations, declarative gate operations, continuous assignment operations, user-defined primitive operations, implicit wired operations, delay path operations, system task operations, and system service operations.

6.-8. (Cancelled).

9. (Previously Presented) The method of claim 23, wherein said step of converting comprises the steps of translating the HDL coded electronic circuit model portions into an intermediate program language code and compiling the intermediate program language code to said simulator-independent binary object code.

10. (Original) The method of claim 9, wherein said intermediate program language code is a C program language code.

11. (Previously Presented) The method of claim 10, wherein said step of converting further comprises the step of grouping said C program language code into types selected from a group consisting of: evaluation C code and scheduling C code.

12. (Previously Presented) The method of claim 23, wherein said simulator-independent binary object code performs operations that are selected from a group consisting of: initial/always block operations, timing-free procedural operations, task procedural operations,

Application No. 09/668,109

function procedural operations, event control operations, delay control operations, scheduled procedural operations, declarative gate operations, continuous assignment operations, user-defined primitive operations, implicit wired operations, delay path operations, system task operations, and system service operations.

13.-15 (Cancelled).

16. (Previously Presented) The system of claim 24, wherein said simulator-independent program comprises binary object code.

17. (Previously Presented) The system of claim 24, wherein said conversion means includes means for translating the HDL coded electronic circuit model portions into an intermediate program language code and means for compiling said intermediate program language code to said simulator-independent program.

18. (Original) The system of claim 17, wherein said intermediate program language code is a C program language code.

19. (Original) The system of claim 18, wherein said C program language code is grouped into code types selected from a group consisting of: evaluation C code and scheduling C code.

Application No. 09/668,109

20. (Previously Presented) The system of claim 24, wherein said simulator-independent program performs operations that are selected from a group consisting of: initial/always block operations, timing-free procedural operations, task procedural operations, function procedural operations, event control operations, delay control operations, scheduled procedural operations, declarative gate operations, continuous assignment operations, user-defined primitive operations, implicit wired operations, delay path operations, system task operations, and system service operations.

21. (Cancelled).

22. (Previously Presented) A simulator-independent, model compiler system for compiling an electronic circuit model, wherein said electronic circuit model comprises a plurality of electronic circuit model portions, wherein said portions have been coded into a hardware description language (HDL), and wherein at least one portion has been coded into a hardware description language that is different from the hardware description language of another portion, said system comprising:

a processor having memory for storing a program that is capable of being executed by said processor, said program directing the operation of the processor to:

convert each of the HDL coded electronic circuit model portions to a simulator-independent binary object code, wherein each binary code utilizes a similarly configured simulator-independent binary object code sequence regardless of the HDL used to code said electronic circuit model portion; and

Application No. 09/668,109

enable said simulator-independent object code to make calls to a programming language interface (PLI) of a simulator regardless of the HDL of said simulator.

23. (Previously Presented) A method for compiling an electronic circuit model, wherein said electronic circuit model comprises a plurality of electronic circuit model portions, wherein said portions have been coded into a hardware description language (HDL), and wherein at least one portion has been coded into an HDL that is different from the HDL of another portion, said method comprising the steps of:

reading the HDL coded electronic circuit model portions;

converting each of the HDL coded electronic circuit model portions to a simulator-independent binary object code, wherein each binary object code utilizes a singularly configured simulator-independent binary object code sequence regardless of the HDL used to code said electronic circuit model portion; and

enabling said simulator-independent object code to make calls to a programming language interface (PLI) of a simulator regardless of the HDL of said simulator.

24. (Previously Presented) A system for compiling an electronic circuit model, wherein said electronic circuit model comprises a plurality of electronic circuit model portions, wherein said portions have been coded into a hardware description language (HDL), and wherein at least one portion has been coded into an HDL that is different from the HDL of another portion, said system comprising:

Application No. 09/668,109

processing means for executing a program, wherein said program includes a conversion means for converting the HDL coded electronic circuit model portions into a singularly configured simulator-independent program, wherein said means for converting includes means for enabling calls to a programming language interface (PLI) of a simulator from said simulator-independent program regardless of the HDL utilized by said simulator.